

Blatt 3 – Zeichnen für Profis

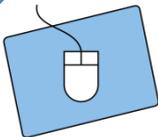


Bisher habt ihr euch um die Malfläche und das Hintergrundbild eurer Zeichenapp gekümmert. Jetzt stehen das **Aussehen** der App sowie das eigentliche **Zeichnen** auf dem Plan. Konkret werdet ihr ...

- **Screen Arrangements** (Anordnungen) einsetzen, um Komponenten einer App auf dem Display anzuordnen.
- dafür sorgen, dass man mit **verschiedenen Farben** auf dem Display zeichnen kann.

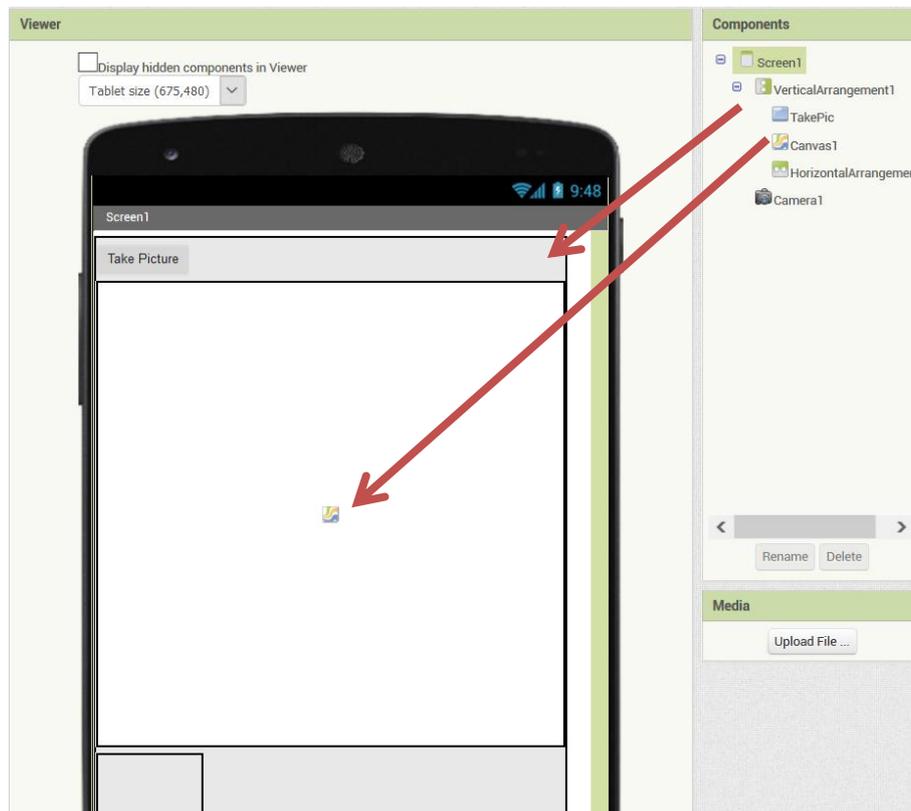


Falls ihr nicht mehr wisst, wie die Screen Arrangements funktionieren, fragt bei einem Betreuer oder einer Betreuerin nach.



Ihr seid dran:

- 1.) Zieht ein **vertikales Screen Arrangement** im Viewer in die App.
- 2.) Zieht den **Aufnahme-Button** und die **Leinwand** in das Arrangement.
- 3.) Zieht ein **horizontales Arrangement** in das **vertikale Arrangement** hinein.

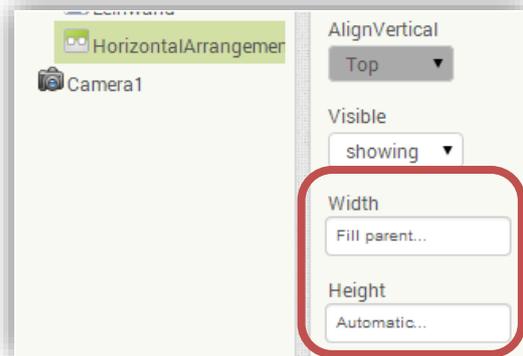


In das jetzt noch leere horizontale Arrangement werden gleich die Elemente für das Zeichnen eingebaut. Zieht zunächst den **Aufnahme-Button** in das horizontale Arrangement.

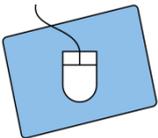
Blatt 3 – Zeichnen für Profis



Um den Darstellungsbereich eures Tablets voll auszunutzen, könnt ihr die **Breite (Width)** der beiden Screen Arrangements auf **Fill Parent** setzen.



Jetzt wird endlich gezeichnet! Erst ein Punkt...



1.) Sucht euch im **Blocks-Editor** aus dem Menü für die Leinwand diesen Block:

```
when Leinwand .TouchDown
  x y
do
```

- 2.) In die Lücke gehört der lila Block rechts, den ihr ebenfalls unter Leinwand findet. Er zeichnet einen Punkt (*draw Point – zeichne Punkt*) an die Koordinaten, die bei x und y stehen.
- 3.) Fehlt noch der Platz, an dem der Punkt erscheinen soll. Fahrt mit der Maus über x bzw. y, und zieht die **get-Blöcke** in die entsprechende Lücke. Diese liefern euch die Koordinaten der Stelle, auf die ihr gedrückt habt.

```
call Leinwand .DrawPoint
  x
  y
```

```
when Leinwand .TouchDown
  x y
do
  call Leinwand .DrawPoint
    x get x
    y get y
```



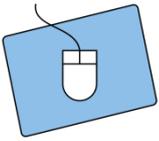
Zwischenergebnis 4

Euer Designeditor sollte nun ungefähr so aussehen. Wenn ihr in der App auf die Leinwand klickt, dann sollte dort ein schwarzer Punkt erscheinen. Ist das nicht der Fall, spricht einen Betreuer oder eine Betreuerin an. Gut gemacht! Jetzt werdet ihr noch unterschiedliche Farben einfügen.



Blatt 3 – Zeichnen für Profis

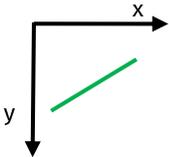
...dann ein Strich!



Ähnlich wie bei einem Punkt funktioniert das auch bei einer Linie: Sucht euch die folgenden Blöcke aus dem **Leinwand-Menü**, und setzt sie sinnvoll zusammen:

```

when Leinwand .Dragged
  startX startY prevX prevY currentX currentY draggedSprite
do
  call Leinwand .DrawLine
    x1
    y1
    x2
    y2
  
```



Der braune Block legt fest, was passiert, wenn man den Finger über den Bildschirm zieht (*drag* – ziehen): Der lila Block sorgt dafür, dass eine Linie zwischen den **Punkten (x1, y1) und (x2, y2)** gezeichnet wird (*draw line* – Linie zeichnen). In die Lücken für x2 und y2 gehören die get-Blöcke aus **currentX** bzw. **currentY** (*current* – aktuell).

Welches Koordinatenpaar (start oder prev) gehört in die (x1, y1)-Lücken?
Was passiert, wenn man die falschen Blöcke wählt?



- × **startX** und **startY** liefern den Punkt, an dem der Bildschirm **zuerst berührt** wurde.
- × **prevX** und **prevY** liefern den Punkt, an dem der Finger **vorher** (previous – vorher) auf dem Bildschirm war.

Einfach mal testen!

```

when Leinwand .Dragged
  startX startY prevX prevY currentX currentY draggedSprite
do
  call Leinwand .DrawLine
    x1
    y1
    x2
    y2
  
```

The code block is annotated with red circles and arrows pointing to the x1, y1, x2, and y2 slots. The x1 and y1 slots are circled in red, with arrows pointing to 'get startX' and 'get startY' blocks respectively. The x2 and y2 slots are also circled in red, with arrows pointing to 'get prevX' and 'get prevY' blocks respectively. Question marks are placed above the 'get startX' and 'get prevY' blocks.



Zwischenergebnis 5

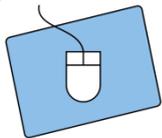
Eure App sollte immer noch so aussehen wie vorher. Wenn ihr jetzt allerdings auf den Bildschirm tippt, solltet ihr einen Punkt malen. Wenn ihr über eure Bildschirm wischt, sollte eine Linie entstehen, die eurem Finger folgt.

Achtung: Wenn ihr immer Linien zwischen dem Startpunkt und eurem Finger malt, hat sich ein Fehler eingeschlichen. Fragt für Hilfe einen Betreuenden.



Alles wird bunt!

Damit ihr mit eurer App auch farbig zeichnen könnt, sorgt ihr jetzt dafür, dass man zwischen verschiedenen **Farben** und **Strichstärken** wählen kann.



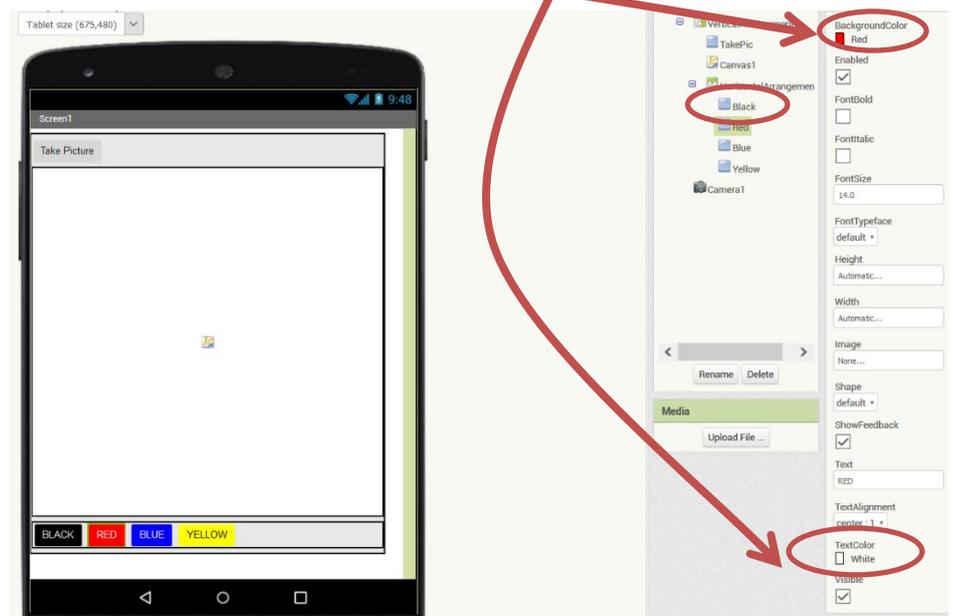
1. Wechselt in den **Designer**, und zieht vier **Buttons** in das leere horizontale Arrangement. Sie sollen zur Farbauswahl eingesetzt werden.
2. Benutzt **Rename**, um die Buttons in **Schwarz**, **Rot**, **Blau** und **Gelb** umzubenennen, und ändert anschließend auch den Text der Buttons sinnvoll. Ihr könnt natürlich auch andere Farben verwenden.
3. Nutzt die **Properties**, um die Buttons zu gestalten. Ihr könnt die Funktionen **BackgroundColor** (Hintergrundfarbe) oder **TextColor** (Textfarbe) nutzen.



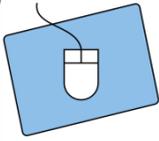
Zwischenergebnis 6

6

So könnte das Ganze dann aussehen:



Blatt 3 – Zeichnen für Profis



- 1.) Wechselt in den **Blocks-Editor**. Sucht euch dort einen **Farb-Button**, und zieht den Click-Block auf die Arbeitsfläche:

```
when Rot .Click
do
```

- 2.) Im Auswahlmenü eurer **Leinwand** findet ihr diesen Block:

```
set Leinwand . PaintColor to
```

Er gehört in die Lücke des Click-Blocks.

- 3.) Unter **Colors** findet ihr Blöcke für die Farben. Zieht die passende Farbe in die hintere Lücke:

```
when Rot .Click
do set Leinwand . PaintColor to
```

- 4.) Wiederholt das Ganze für die übrigen Farben.



Zwischenergebnis 7

Eure App sollte jetzt immer noch so aussehen wie in Zwischenergebnis 6.

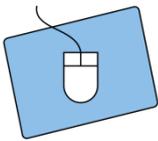
Allerdings sollte sich nun die Farbe, in der die Punkte gemalt werden, ändern, wenn ihr auf den entsprechenden Button drückt. Dabei müsst ihr nicht unbedingt die gleichen Farben haben wie wir im Beispiel.

Der Foto-aufnehmen-Button sollte über der Leinwand sein, die anderen Buttons unter der Leinwand (oder umgekehrt, das überlassen wir euch).

Solltet ihr noch irgendwelche Probleme haben, spricht eine Betreuerin oder einen Betreuer an.

Blatt 3 – Zeichnen für Profis

Stiftbreite und Wischer



Um die Stiftbreite einzustellen und einen Wischer/Radiergummi einzubauen, wechselt ihr wieder in den Designer.

- 1.) Zieht ein **neues horizontales Screen Arrangement** in die App, in das die neuen Elemente eingebaut werden.
- 2.) Zieht zwei **Labels** (Textfelder) aus **User Interface** in das neue Arrangement.
- 3.) Gebt dem ersten Label sowohl den Namen als auch den Text „Stiftbreite“. Das zweite Label erhält den Namen „Wert_Stiftbreite“ und als Text eine 2 (Voreinstellung für die Stiftbreite).
- 4.) Als Nächstes benötigt ihr **zwei Buttons**, die für die Einstellung der **Stiftbreite** zuständig sind. Gebt ihnen passende Namen, z. B. „Breite_rauf“ und „Breite_runter“, und ändert ihre Texte sinnvoll.



- 5.) Fehlt noch der **Wischer**, der die Leinwand abwischt. Hier benötigt ihr noch einen **Button**, der eine passende **Bezeichnung** und eine **Beschriftung** bekommt.
- 6.) **Wechselt in den Blocks-Editor**. Für den Wischer braucht ihr die folgenden Blöcke:



Könnt ihr euch denken, wo ihr die herbekommt?

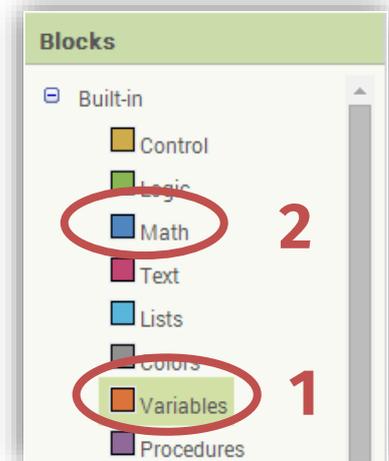
- 7.) Damit der Stift (euer Finger) malen kann, müsst ihr ihm eine Stiftbreite zuweisen. Dazu legt ihr eine **Variable** (siehe 1) an und nennt sie „Stiftbreite“:



Hier klicken, um den Namen zu ändern.

Hier fehlt noch der Startwert.

- 8.) Den Startwert (die voreingestellte Stiftbreite) legt ihr fest, indem ihr den Zahlen-Baustein aus Math (siehe 2) in die Lücke einbaut und die 2 einträgt. Das sieht dann so aus:

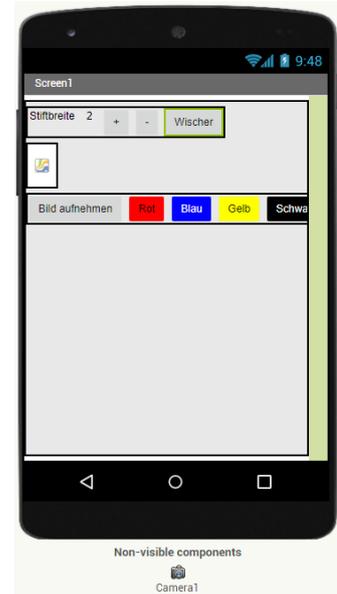




Zwischenergebnis 8

Eure App sollte jetzt wie auf dem Bild aussehen.

Wenn ihr auf den Wischer drückt, verschwindet alles, was ihr zuvor auf die Leinwand gemalt habt. Sollten euch Unterschiede auffallen, könnt ihr einen Betreuenden auf euch aufmerksam machen. Wenn alles stimmt, geht weiter, und ihr werdet die Stiftbreite, mit der die Punkte auf die Leinwand gebracht werden, verändern!



Die Stiftbreite einstellen

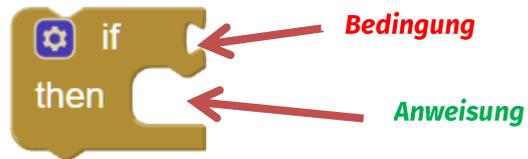
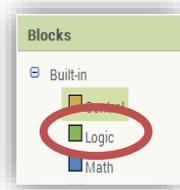
Die Stiftbreite sollte sich in einem Bereich von 1 bis 5 Pixeln (Bildpunkt) bewegen. Also müsst ihr testen, ob sich die Stiftbreite in diesem Bereich befindet, bevor ihr sie ändert. Dafür nutzt ihr die **if-then-Anweisung** (Wenn-dann-Anweisung), die dafür sorgt, dass die Ausführung eines Programnteils nur dann erfolgt, wenn eine bestimmte Bedingung erfüllt ist. Ein Beispiel:

Wenn morgen die Sonne scheint, dann gehe ich ins Freibad.

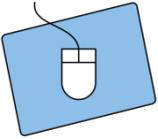
↑
Bedingung

↑
Anweisung

Im App Inventor ist hierfür der **if-then-Block** zuständig, den ihr im Blocks-Editor unter **Control** findet.



Blatt 3 – Zeichnen für Profis

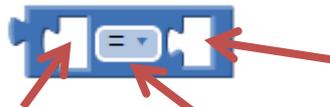


Die Stiftbreite erhöhen:

- Sucht euch den **Click-Block** für den Button, der die Stiftbreite **erhöhen** soll. In die Lücke baut ihr den **if-then-Block** aus **Control** ein.



- Für die **Bedingung** müsst ihr hier testen, ob die Stiftbreite noch **kleiner als** der Maximalwert ist. Ihr braucht also einen mathematischen Vergleich aus **Math**:



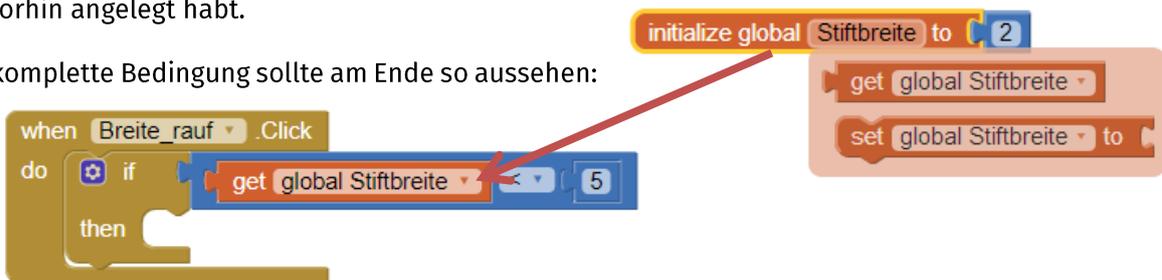
Hierhin gehört die aktuelle Stiftbreite.

Hier steht der Maximalwert, also die **Zahl 5**.

Hier kann der Vergleichsoperator ausgewählt werden (ihr benötigt <).

Die aktuelle Stiftbreite für den Vergleich erhaltet ihr über den **get-Block** aus der globalen Variablen, die ihr vorhin angelegt habt.

Die komplette Bedingung sollte am Ende so aussehen:



Im **Then**-Teil sind die folgenden Schritte zu erledigen:

- Die Stiftbreite um 1 erhöhen:



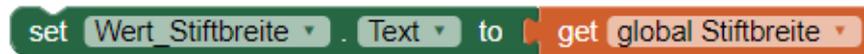
Den **set-** und **get-Block** erhaltet ihr wieder aus der globalen Variablen, den Block für die **Addition** und den für die **Zahl** aus **Math**.

- Die Stiftbreite an die Leinwand weitersagen:



Den grünen Block findet ihr unter **Leinwand**. Er setzt die Linienbreite (Linewidth) der Leinwand auf den Wert, der hinten angehängen wird.

- Das **Label** mit der Stiftbreite aktualisieren:



Den **set-Block** gibt euch das Menü für euer Label **Wert_Stiftbreite**.

Blatt 3 – Zeichnen für Profis

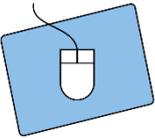


Zwischenergebnis 9

Wenn ihr mit allen Schritten fertig seid, sollte euer Block zum Erhöhen der Stiftbreite so aussehen:

```

when Breite_rauf .Click
do
  if get global Stiftbreite < 5
  then
    set global Stiftbreite to get global Stiftbreite + 1
    set Leinwand . LineWidth to get global Stiftbreite
    set Wert_Stiftbreite . Text to get global Stiftbreite
  
```



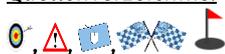
Setzt nun genauso den Block zum **Verringern** der Stiftbreite zusammen. Allerdings müsst ihr hier testen, ob der Wert der Stiftbreite noch **größer als das Minimum** ist, und – wenn ja – die Stiftbreite um 1 verringern.

Wenn ihr euer letztes Todo auf dieser Seite erledigt habt, sollte eure App nun vollständig funktionieren ☺.

Super! Damit ist eure Zeichenapp fertig. Ihr könnt sie jetzt testen und noch das Layout optimieren. Wenn ihr zufrieden seid, könnt ihr anfangen, ein Spiel zu programmieren! Meldet euch einfach bei den Betreuerinnen und Betreuern.



Quellenverzeichnis:



– Quelle: InfoSphere

Alle weiteren Abbildungen – Quelle: Screenshots des MIT-App-Inventors (<http://appinventor.mit.edu>)